

**Amendments to the Specification**

Please replace paragraph [0003] with the following amended paragraph:

[0003] While object oriented programming has several advantages, it can be a difficult and time consuming process for programmers to develop complex object oriented applications. The programmer must first determine the parameters of the application desired (e.g., input variable, desired screen appearance, output variables, etc). An appropriate format or language is then selected to create the application (e.g., ~~[[g]]Gif~~, Java, J2EE, C/C++, etc.). Then the programmer must draft the programming code using the desired format in order to create the application or desired objects. This can be a time intensive, detailed process.

Please replace paragraph [0005] with the following amended paragraph:

[0005] Programming models have been developed using XML technologies in an effort to assist programmers in developing object oriented programs. Programming models include XML templates and eXtensible Stylesheet Language (XSL) models. XSL models transform XML documents into stylized versions of the original document. While an XSL style sheet can be developed to assist in creating a specific application, significant programming resources are necessary to develop a particular model for a particular application.

Please replace paragraph [0006] with the following amended paragraph:

[0006] The present invention provides a method, system, and computer program product for automatically developing objects using a plurality of context derived models (e.g., XML, ~~[[g]]Gif~~, Java, C/C++) residing within a computational grid. An object meta language (OML) is used to allow a programmer to define an application. Using OML, the programmer creates a document describing the required object. The OML document is submitted to a group of context derived models residing at various computational nodes on the grid. A web service is used to parse the OML document and select the appropriate node. The OML document is provided to the selected node, which applies object description variables using a transform language, such as eXtensible Stylesheet Language (XSL). The defined object is then returned to the programmer, thus eliminating the need for the programmer to generate the actual code for the desired object.

Please replace paragraph [0012] with the following amended paragraph:

[0012] Each computational grid 15, 17 comprises a series of nodes 16a, 16b, 16c, 16d, 16e, 16f, 18a, 18b, 18c, 18d, 18e, 18f. Each node contains one or more programming models. In the exemplary implementation, the programming models comprise ~~XML templates~~ XSL style sheets (16a-18f) and ~~XSL style sheets~~ XML template (18a-18f). Models constructed using other formats could also be used.

Please replace paragraph [0015] with the following amended paragraph:

[0015] At step 23, the OML document is provided to a web service. The web service parses the document using XML technologies to determine a suitable code generation module. At step 25, the web service surveys all available grid nodes in order to locate an available node that comprises a XSL style sheet or XML template in accordance with the application parameters set forth in the OML document. In an exemplary embodiment, the grid nodes contain XSL style sheets or XML templates capable of generating completely coded applications from XML definitions. Alternatively, the nodes could contain simple Gif generation modules or more complex applications such as Java applications, J2EE applications, or C/C++ applications. The web service selects the appropriate module based upon the OML definitions set forth by the author.

Please replace paragraph [0016] with the following amended paragraph:

[0016] Once a suitable style sheet or template is located, the web service provides the parsed OML definition to the selected node (step 27). The node on the grid uses a context derived model to generate the code for the desired application. For example, a particular node may contain an XSL style sheet to perform an XSL/XML transform. An XML/XSL transform is used to take XML application definitions and create a fully described application by applying a predetermined style template to the parsed XML code (step 29). For example, an author might express a desire for the application to display a title by using an XML <title> tag. The XSL module takes this information and creates a styled title definition (e.g., a particular bold faced font, a shaped word configuration, a word/motion combination). The models residing on the various nodes within the grid are able to create various output formats, depending upon the application desired. For example, after the OML definition is parsed by the node and the XML/XSL transform is applied, the output from the XML/XSL transform can be an XML document, a Java Server Page (JSP), a Java application, etc.

Please replace paragraph [0018] with the following amended paragraph:

[0018] To clearly explain the manner in which the present invention operates, an example of an author's OML application definition is shown below as Example #1:

**EXAMPLE #1**

```
1 <?XML version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE oml-def PUBLIC "-//IBM//OML DTD 1.0//EN" "oml_1.0.dtd">
3 <object_context com="IBM" prer="IBM WPS 4.1.4">
  .
  .
  .
4 <oml-title>The process box example</oml-title>
5 <oml-subobject graphic=button gstyle_title="Start" action_object=URLGet>
```

*More object description OML*

**6 </object\_context>**

Please replace paragraph [0024] with the following amended paragraph:

[0024] The XSL document reads the OML variables set forth by the author and substitutes them into new objects via the XSL syntax. In the example #2, the model is defined as an XML version 1.0 type document that is an XSL style\_sheet using XSL transformation specification level 1.0 from the w3.org 1999 specification (lines 1-4). These designated specifications are well known within the art.


Please replace paragraph [0026] with the following amended paragraph:

[0026] The XSL style\_sheet of example #2 will parse the input OML file and substitute the value of the variable set in the input document with the value of the file variables in the template. For example, the commands in the XSL style\_sheet check for a match of the OML variable "title" and replace it with the XSL element for the title variable using an "if" clause (lines 10-15 of example #2). In this manner, the XSL style\_sheet is used to replace the variable set from the input OML document with the variable set that is the result of the XSL/XML transform. In other words, the instructions within the XSL style sheet substitute the various variables found in the XML application definition into the styled format which is created by the template.

Please replace paragraph [0027] with the following amended paragraph:

[0027] Once OML description has been transformed into a fully coded application using the XSL style\_sheet, it is output to an output file (in the example, to a java file). This file is then returned to the programmer. In the exemplary embodiment, the file is returned via the web service (e.g., WebSpere Portal Server 4.1.4).

Please replace paragraph [0029] with the following amended paragraph:

[0029] The system in accordance with the present invention allows for programmers to develop applications without tedious coding. Virtually any type of object can be wrapped in an OML format that can then easily be parsed, understood, and modified using existing templates and style sheets. As a result, programming time and costs are reduced. Additionally, the level of programming skill required to be possessed by the author is significantly reduced. No longer must the author be proficient in all of the languages or code scripts desired. As long as the author can describe the application via OML text, the system in accordance with the present invention will develop the actual coded application. Using this method, any programming object from a simple  file to a complex three tier J2EE application can be generated from existing data models.